AobaZero の 2023 年のアピール文書

山下 宏
yss@bd.mbn.or.jp

1 AlphaZero の追試が最初の目的

AobaZero は Bonanza、LeelaZero のコードをベースに AlphaZero の追試をするべく MCTS +ディープラーニングで実装されてます。ネットワークは 3x3 のフィルタが 256 個の 20 block の ResNet でパラメータの個数は 2340 万個。棋譜生成をユーザの皆様と協力して行う分散強化学習です。オープンソースです[*1]。

2 AlphaZero の追試は 2021 年 4 月に終了

AlphaZero の将棋の追試は、2019 年 3 月から開始し、2021 年 4 月に 3900 万棋譜を作成して終了しました[*2]。2023 年 3 月 31 日現在、6363 万棋譜を作成しています。

3 追試終了後から +230 ELO、去年の選手権から +100 ELO

追試終了後からは +230 ELO、去年の選手権からだと +100 ELO ほど強くなっています。効果があった主な変更は

- 3 手詰を全ノードで、dfpn の詰を全ノードで。+40 ELO(10visit で 10000 ノード探索、100visit で 10 万ノード、と 1000 倍のノード数で)
- Root の Policy の Softmax 温度を 1.0 から 1.8 に。+50 ELO
- 序盤や早い投了棋譜の学習割合を減らす。自己対戦を 1 手平均 800 から 1600 playout に。+60 ELO
- UCB 値を計算するときの定数、cPUCT の値を動的に変更。+25 ELO。playout ごとの評価値の分散を求めて大きい場合は cPUCT の値も大きく[*3]。KataGo で使われている Dynamic Variance-Scaled cPUCT[*4]。
- 互角の局面の学習確率を減らし +28 ELO。

追試終了時では AlphaZero より +150 ELO 弱い、という推定でしたが現在は +230 なので AlphaZero を +80 ELO 程度、超えた棋力かもしれません。

[*1] https://github.com/kobanium/aobazero
[*2] https://github.com/kobanium/aobazero/issues/54
[*3] http://www.yss-aya.com/bbs/patio.cgi?read=33&ukey=0
[*4] https://github.com/lightvector/KataGo/blob/master/docs/KataGoMethods.md#dynamic-variance-scaled-cpuct

4 互角の局面の学習確率を減らす

最後の手法について詳しく書いてみます。おそらくこれは新規の手法だと思います。AlphaZero は生成した棋譜の全局面を同じ確率で学習に使用します。AobaZero も同じやり方でした。過去 100 万棋譜の Replay Buffer に含まれる、約 9000 万局面からランダムに 128 局面を選び、ミニバッチを作成して学習を繰り返します。これだと平手の初期局面は 9000 万中、100 万局面も含まれるため、0 手目から 30 手目までの選択確率を減らしていました。

今回、さらに局面の勝率が互角近い (勝率 0.50) 局面の学習確率を減らし、勝率 0.30 や勝率 0.70 と形勢に差がついた局面の確率を上げています。具体的には局面の選択確率を

- 勝率 0.50-0.60 は 1 倍 (勝率 0.50-0.40 も、以下同)
- 勝率 0.60-0.70 は 2 倍
- 勝率 0.70-0.80 は 13 倍
- 勝率 0.80-1.00 は 8 倍

としてランダム初期値から学習しなおした結果、現在の重みよりも +28 ELO 強いものが出来ました。

4.1 実験結果

表 1 は対水匠 5 での AobaZero から見た ELO 差です。

表 1 対水匠 5 の ELO 差

1 手の playout 数	ELO	差
100 (基準)	53	
100 互角局面の割合を減らす	70	+17
800 (基準)	40	
800 互角局面の割合を減らす	68	+28

2400 局ずつ。水匠 5(7.50,1 手 40k(250k)) と 1 手 100playout(800 playout) の ELO 差。基準は w4195。互角局面集 (24 手目まで)[*5]利用。

4.2 学習のさせ方

ランダム初期値から再学習させた内容は

- 4300 万棋譜から 6325 万棋譜まで ReplayBuffer 300 万棋譜で 160 万回学習。cos annealing で 0.01 から 0.0001

[*5] https://yaneuraou.yaneu.com/2016/08/24/

まで

- 5945 万棋譜から 6344 万棋譜まで ReplayBuffer 400 万棋譜で 80 万回学習。cos annealing で 0.0001 から 0.000002 まで
- ミニバッチ 256。合計 6 億局面。

パラメータ調整の実験は 192x10block のネットで初期値から 1000 万局面を学習させたものに、条件を変えてさらに 1000 万局面を学習で行いました。実験だと自己対戦だと互角局面を減らす方が +100ELO ほど、対水匠でも +50ELO ほど強くなるのですが学習回数を増やすとだんだん効果は下がっていくようです。

学習の loss は Policy の交差エントロピーと Value(対局結果と局面の勝率の平均) の二乗誤差、重みの L2 正則化です。

4.3 何手目を学習させているか

図 1 は手数により学習される局面の割合です。左が AlphaZero の方式で、中央が 30 手目までを減らしたもの、右が互角を減らした場合です。200 手以上は累積です。互角の局面を減らすことで手数のピークがほぼなくなり、30 手目から 100 手目まで均等な割合で学習しています。

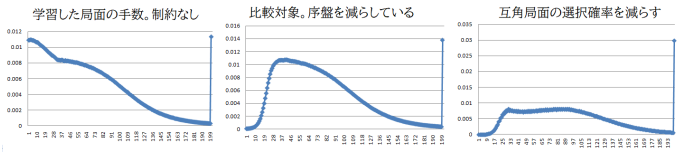

図 1　棋力の推移。右軸が floodgate のレート、横軸は棋譜数 (万)

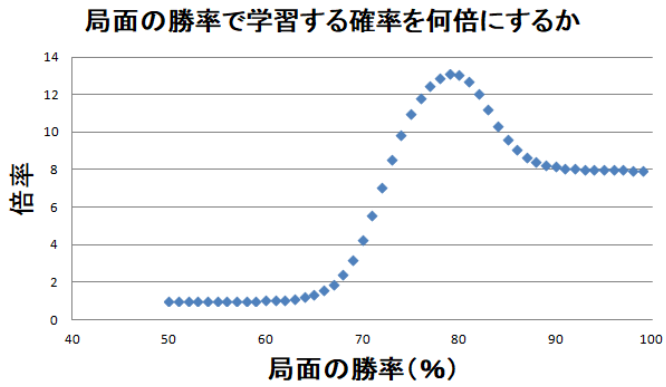

図 2　局面の勝率で学習する確率を何倍にするか

図 2 は勝率により学習される確率を何倍にしてるか、です。最初に書いた大雑把なやり方 (0.7 で 13 倍) でもほぼ結果は一緒ですが実際は適当な関数で近似してます[*6]。

[*6] https://github.com/kobanium/aobazero/blob/develop/learn/yss_dcnn.cpp#L3535

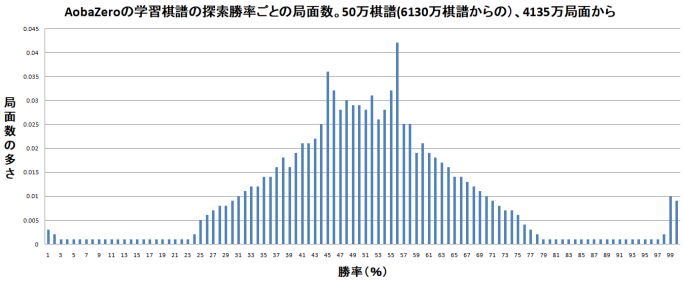

図 3　探索勝率ごとの局面数

図 3 は局面ごとの勝率の分布です。勝率 25 %以下、75 %以上が少ないのは現在の投了の閾値が 0.25 なためです。56 %と 45 %にピークがあるのは初手 (56 %) と 2 手目の勝率 (45 %) のためです。棋譜の 10 %は投了禁止で最後まで指しています。

なぜこの手法で棋力が上がるのかは分かりません。勝率に差がついた局面が正確になる方が、探索中に勝率に差がつくことに敏感に反応できるせいかもしれません。もしくは未知の局面の勝率を 0.5 と推測するのは容易なのかもしれません。

4.4 実装

100 万個以上から選択確率が異なるものをランダムに選ぶのは難しいのですが、Sum-Tree という 2 分木を利用した手法が知られており、これを利用しました。山岡さんの記事[*7]を参考にしています。AobaZero での実装はこちら[*8]です。

5 人間の知識は使っていない、をおそらく継続

利きの情報の追加や 3 手詰、dfpn 詰などで AlphaZero からは離れてきましたが、まだ全体としては「人間の知識は使っていない」を継続していると考えています。

6 棋力の推移

図 4 が ELO の推移です。floodgate での測定レートの方が若干高めなのは Kristallweizen での棋力測定に用いている互角局面集 (24 手まで) には AobaZero が指さない穴熊や振飛車が多数含まれているせいです。局面集を使わずに初手から指させた方が +100 ELO ほど強くなります。

7 4 年で 6300 万棋譜

6300 万棋譜、という膨大な棋譜を 4 年間で生成してきました。棋譜生成に協力していただいてる皆様に感謝いたし

[*7] https://tadaoyamaoka.hatenablog.com/entry/2019/08/18/154610

[*8] https://github.com/kobanium/aobazero/blob/develop/learn/yss_dcnn.cpp#L2809

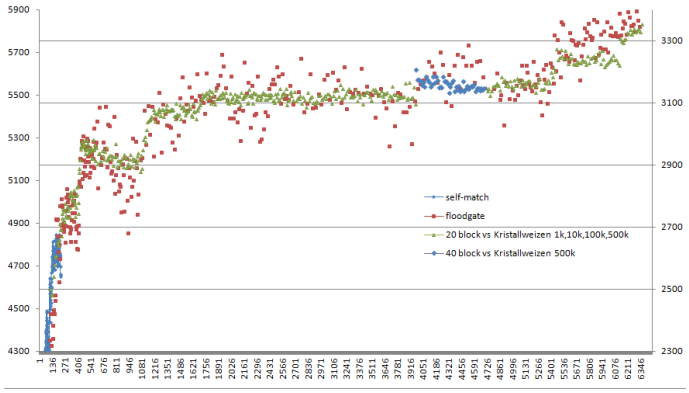

図 4　棋力の推移。右軸が floodgate のレート、横軸は棋譜数 (万)

ます。